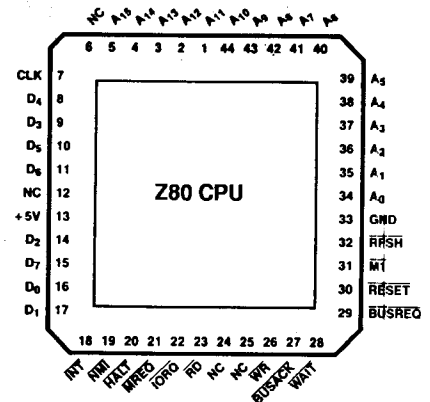


**44 pin Quad Flat Pack (QFP), Pin Assignments**  
(Only available for 84C00)

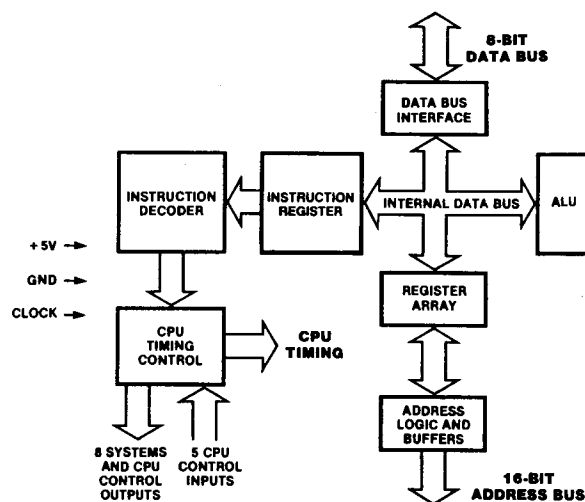


**Figure 2b. 44-Pin Chip Carrier Pin Assignments**

## GENERAL DESCRIPTION

The CPUs are fourth-generation enhanced microprocessors with exceptional computational power. They offer higher system throughput and more efficient memory utilization than comparable second- and third-generation microprocessors. The internal registers contain 208 bits of read/write memory that are accessible to the programmer. These registers include two sets of six general-purpose registers which may be used individually as either 8-bit registers or as 16-bit register pairs. In addition, there are two sets of accumulator and flag registers. A group of "Exchange" instructions makes either set of main or alternate registers accessible to the programmer. The alternate set allows operation in foreground-background mode or it may be reserved for very fast interrupt response.

The CPU also contains a Stack Pointer, Program Counter, two index registers, a Refresh register (counter), and an Interrupt register. The CPU is easy to incorporate into a system since it requires only a single +5V power source. All output signals are fully decoded and timed to control standard memory or peripheral circuits; the CPU is supported by an extensive family of peripheral controllers. The internal block diagram (Figure 3) shows the primary functions of the processors. Subsequent text provides more detail on the I/O controller family, registers, instruction set, interrupts and daisy chaining, and CPU timing.



**Figure 3. Z80C CPU Block Diagram**

Table 1. Z80C CPU Registers

Register		Size (Bits)	Remarks
A, A'	Accumulator	8	Stores an operand or the results of an operation.
F, F'	Flags	8	See Instruction Set.
B, B'	General Purpose	8	Can be used separately or as a 16-bit register with C.
C, C'	General Purpose	8	Can be used separately or as a 16-bit register with B.
D, D'	General Purpose	8	Can be used separately or as a 16-bit register with E.
E, E'	General Purpose	8	Can be used separately or as a 16-bit register with D.
H, H'	General Purpose	8	Can be used separately or as a 16-bit register with L.
L, L'	General Purpose	8	Can be used separately or as a 16-bit register with H.
Note: The (B,C), (D,E), and (H,L) sets are combined as follows:			
B — High byte    C — Low byte			
D — High byte    E — Low byte			
H — High byte    L — Low byte			
I	Interrupt Register	8	Stores upper eight bits of memory address for vectored interrupt processing.
R	Refresh Register	8	Provides user-transparent dynamic memory refresh. Automatically incremented and placed on the address bus during each instruction fetch cycle.
IX	Index Register	16	Used for indexed addressing.
IY	Index Register	16	Used for indexed addressing.
SP	Stack Pointer	16	Holds address of the top of the stack. See Push or Pop in instruction set.
PC	Program Counter	16	Holds address of next instruction.
IFF <sub>1</sub> -IFF <sub>2</sub>	Interrupt Enable	Flip-Flops	Set or reset to indicate interrupt status (see Figure 4).
IMFa-IMFb	Interrupt Mode	Flip-Flops	Reflect Interrupt mode (see Figure 4).

failure has been detected. After recognition of the  $\overline{\text{NMI}}$  signal (providing  $\overline{\text{BUSREQ}}$  is not active), the CPU jumps to restart location 0066H. Normally, software starting at this address contains the interrupt service routine.

**Maskable Interrupt ( $\overline{\text{INT}}$ ).** Regardless of the interrupt mode set by the user, the CPU response to a maskable interrupt input follows a common timing cycle. After the interrupt has been detected by the CPU (provided that interrupts are enabled and  $\overline{\text{BUSREQ}}$  is not active) a special interrupt processing cycle begins. This is a special fetch ( $\overline{\text{M1}}$ ) cycle in which  $\overline{\text{IORQ}}$  becomes active rather than  $\overline{\text{MREQ}}$ , as in a normal  $\overline{\text{M1}}$  cycle. In addition, this special  $\overline{\text{M1}}$  cycle is automatically extended by two  $\overline{\text{WAIT}}$  states, to allow for the time required to acknowledge the interrupt request.

**Mode 0 Interrupt Operation.** This mode is similar to the 8080 microprocessor interrupt service procedures. The interrupting device places an instruction on the data bus. This is normally a Restart instruction, which will initiate a call

to the selected one of eight restart locations in page zero of memory. Unlike the 8080, the Z80 CPU responds to the Call instruction with only one interrupt acknowledge cycle followed by two memory read cycles.

**Mode 1 Interrupt Operation.** Mode 1 operation is very similar to that for the NMI. The principal difference is that the Mode 1 interrupt has only one restart location, 003BH.

**Mode 2 Interrupt Operation.** This interrupt mode has been designed to most effectively utilize the capabilities of the Z80 microprocessor and its associated peripheral family. The interrupting peripheral device selects the starting address of the interrupt service routine. It does this by placing an 8-bit vector on the data bus during the interrupt acknowledge cycle. The CPU forms a pointer using this byte as the lower 8 bits and the contents of the I register as the upper 8 bits. This points to an entry in a table of addresses for interrupt service routines. The CPU then jumps to the routine at that

address. This flexibility in selecting the interrupt service routine address allows the peripheral device to use several different types of service routines. These routines may be located at any available location in memory. Since the interrupting device supplies the low-order byte of the 2-byte vector, bit 0 ( $A_0$ ) must be a zero.

**Interrupt Enable/Disable Operation.** Two flip-flops, IFF<sub>1</sub> and IFF<sub>2</sub>, referred to in the register description, are used to signal the CPU interrupt status. Operation of the two flip-flops is described in Table 2. For more details, refer to the *Z80 CPU Technical Manual* (03-0029-01) and *Z80 Assembly Language Programming Manual* (03-0002-01).

Table 2. State of Flip-Flops

Action	IFF <sub>1</sub>	IFF <sub>2</sub>	Comments
CPU Reset	0	0	Maskable interrupt INT disabled
DI instruction execution	0	0	Maskable interrupt INT disabled
EI instruction execution	1	1	Maskable interrupt INT enabled
LD A,I instruction execution	•	•	IFF <sub>2</sub> → Parity flag
LD A,R instruction execution	•	•	IFF <sub>2</sub> → Parity flag
Accept NMI	0	•	Maskable interrupt INT disabled
RETN instruction execution	IFF <sub>2</sub>	•	IFF <sub>2</sub> → IFF <sub>1</sub> at completion of an NMI service routine.

## INSTRUCTION SET

The microprocessor has one of the most powerful and versatile instruction sets available in any 8-bit microprocessor. It includes such unique operations as a block move for fast, efficient data transfers within memory, or between memory and I/O. It also allows operations on any bit in any location in memory.

The following is a summary of the instruction set which shows the assembly language mnemonic, the operation, the flag status, and gives comments on each instruction. For an explanation of flag notations and symbols for mnemonic tables, see the Symbolic Notations section which follows these tables. The *Z80 CPU Technical Manual* (03-0029-01), the *Programmer's Reference Guide* (03-0012-03), and *Assembly Language Programming Manual* (03-0002-01) contain significantly more details for programming use.

The instructions are divided into the following categories:

- ☐ 8-bit loads
- ☐ 16-bit loads
- ☐ Exchanges, block transfers, and searches
- ☐ 8-bit arithmetic and logic operations
- ☐ General-purpose arithmetic and CPU control
- ☐ 16-bit arithmetic operations
- ☐ Rotates and shifts

- ☐ Bit set, reset, and test operations
- ☐ Jumps
- ☐ Calls, returns, and restarts
- ☐ Input and output operations

A variety of addressing modes are implemented to permit efficient and fast data transfer between various registers, memory locations, and input/output devices. These addressing modes include:

- ☐ Immediate
- ☐ Immediate extended
- ☐ Modified page zero
- ☐ Relative
- ☐ Extended
- ☐ Indexed
- ☐ Register
- ☐ Register indirect
- ☐ Implied
- ☐ Bit

## 8-BIT LOAD GROUP

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 78 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD r, r'	$r \leftarrow r'$	•	•	X	•	X	•	01 r r'		1	1	4	r, r' Reg.
LD r, n	$r \leftarrow n$	•	•	X	•	X	•	00 r 110		2	2	7	000 B
								$\leftarrow n \rightarrow$					001 C
LD r, (HL)	$r \leftarrow (HL)$	•	•	X	•	X	•	01 r 110		1	2	7	010 D
LD r, (IX+d)	$r \leftarrow (IX+d)$	•	•	X	•	X	•	11 011 101	DD	3	5	19	011 E
								01 r 110					100 H
								$\leftarrow d \rightarrow$					101 L
LD r, (IY+d)	$r \leftarrow (IY+d)$	•	•	X	•	X	•	11 111 101	FD	3	5	19	111 A
								01 r 110					
								$\leftarrow d \rightarrow$					
LD (HL), r	$(HL) \leftarrow r$	•	•	X	•	X	•	01 110 r		1	2	7	
LD (IX+d), r	$(IX+d) \leftarrow r$	•	•	X	•	X	•	11 011 101	DD	3	5	19	
								01 110 r					
								$\leftarrow d \rightarrow$					
LD (IY+d), r	$(IY+d) \leftarrow r$	•	•	X	•	X	•	11 111 101	FD	3	5	19	
								01 110 r					
								$\leftarrow d \rightarrow$					
LD (HL), n	$(HL) \leftarrow n$	•	•	X	•	X	•	00 110 110	36	2	3	10	
								$\leftarrow n \rightarrow$					
LD (IX+d), n	$(IX+d) \leftarrow n$	•	•	X	•	X	•	11 011 101	DD	4	5	19	
								00 110 110	36				
								$\leftarrow d \rightarrow$					
								$\leftarrow n \rightarrow$					

## 8-BIT LOAD GROUP (Continued)

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD (IY+d), n	(IY+d) ← n	•	•	X	•	X	•	•	11 111 101 FD 00 110 110 36	4	5	19	
									← d → ← n →				
LD A, (BC)	A ← (BC)	•	•	X	•	X	•	•	00 001 010 0A	1	2	7	
LD A, (DE)	A ← (DE)	•	•	X	•	X	•	•	00 011 010 1A	1	2	7	
LD A, (nn)	A ← (nn)	•	•	X	•	X	•	•	00 111 010 3A	3	4	13	
									← n → ← n →				
LD (BC), A	(BC) ← A	•	•	X	•	X	•	•	00 000 010 02	1	2	7	
LD (DE), A	(DE) ← A	•	•	X	•	X	•	•	00 010 010 12	1	2	7	
LD (nn), A	(nn) ← A	•	•	X	•	X	•	•	00 110 010 32	3	4	13	
									← n → ← n →				
LDA, I	A ← I	‡	‡	X	0	X	IFF	0	• 11 101 101 ED 01 010 111 57	2	2	9	
LDA, R	A ← R	‡	‡	X	0	X	IFF	0	• 11 101 101 ED 01 011 111 5F	2	2	9	
LD I, A	I ← A	•	•	X	•	X	•	•	11 101 101 ED 01 000 111 47	2	2	9	
LD R, A	R ← A	•	•	X	•	X	•	•	11 101 101 ED 01 001 111 4F	2	2	9	

NOTE: IFF, the content of the interrupt enable flip-flop, (IFF<sub>2</sub>), is copied into the P/V flag.

## 16-BIT LOAD GROUP

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD dd, nn	dd ← nn	•	•	X	•	X	•	•	00 dd0 001	3	3	10	dd Pair 00 BC 01 DE
									← n → ← n →				
LD IX, nn	IX ← nn	•	•	X	•	X	•	•	11 011 101 DD 00 100 001 21	4	4	14	10 HL 11 SP
									← n → ← n →				
LD IY, nn	IY ← nn	•	•	X	•	X	•	•	11 111 101 FD 00 100 001 21	4	4	14	
									← n → ← n →				
LD HL, (nn)	H ← (nn+1) L ← (nn)	•	•	X	•	X	•	•	00 101 010 2A	3	5	16	
									← n → ← n →				
LD dd, (nn)	dd <sub>H</sub> ← (nn+1) dd <sub>L</sub> ← (nn)	•	•	X	•	X	•	•	11 101 101 ED 01 dd1 011	4	6	20	
									← n → ← n →				

NOTE: (PAIR)<sub>H</sub>, (PAIR)<sub>L</sub> refer to high order and low order eight bits of the register pair respectively. e.g., BC<sub>L</sub> = C, AF<sub>H</sub> = A.

# 16-BIT LOAD GROUP (Continued)

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD IX, (nn)	$IX_H \leftarrow (nn+1)$ $IX_L \leftarrow (nn)$	•	•	X	•	X	•	•	11 011 101 DD 00 101 010 2A	4	6	20	
									$\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$				
LD IY, (nn)	$IY_H \leftarrow (nn+1)$ $IY_L \leftarrow (nn)$	•	•	X	•	X	•	•	11 111 101 FD 00 101 010 2A	4	6	20	
									$\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$				
LD (nn), HL	$(nn+1) \leftarrow H$ $(nn) \leftarrow L$	•	•	X	•	X	•	•	00 100 010 22	3	5	16	
									$\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$				
LD (nn), dd	$(nn+1) \leftarrow dd_H$ $(nn) \leftarrow dd_L$	•	•	X	•	X	•	•	11 101 101 ED 01 dd0 011	4	6	20	
									$\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$				
LD (nn), IX	$(nn+1) \leftarrow IX_H$ $(nn) \leftarrow IX_L$	•	•	X	•	X	•	•	11 011 101 DD 00 100 010 22	4	6	20	
									$\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$				
LD (nn), IY	$(nn+1) \leftarrow IY_H$ $(nn) \leftarrow IY_L$	•	•	X	•	X	•	•	11 111 101 FD 00 100 010 22	4	6	20	
									$\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$				
LD SP, HL	$SP \leftarrow HL$	•	•	X	•	X	•	•	11 111 001 F9	1	1	6	
LD SP, IX	$SP \leftarrow IX$	•	•	X	•	X	•	•	11 011 101 DD 11 111 001 F9	2	2	10	
LD SP, IY	$SP \leftarrow IY$	•	•	X	•	X	•	•	11 111 101 FD 11 111 001 F9	2	2	10	
PUSH qq	$(SP-2) \leftarrow qq_L$ $(SP-1) \leftarrow qq_H$ $SP \rightarrow SP-2$	•	•	X	•	X	•	•	11 qq0 101	1	3	11	qq Pair
													00 BC 01 DE 10 HL 11 AF
PUSH IX	$(SP-2) \leftarrow IX_L$ $(SP-1) \leftarrow IX_H$ $SP \rightarrow SP-2$	•	•	X	•	X	•	•	11 011 101 DD 11 100 101 E5	2	4	15	
PUSH IY	$(SP-2) \leftarrow IY_L$ $(SP-1) \leftarrow IY_H$ $SP \rightarrow SP-2$	•	•	X	•	X	•	•	11 111 101 FD 11 100 101 E5	2	4	15	
POP qq	$qq_H \leftarrow (SP+1)$ $qq_L \leftarrow (SP)$ $SP \rightarrow SP+2$	•	•	X	•	X	•	•	11 qq0 001	1	3	10	
POP IX	$IX_H \leftarrow (SP+1)$ $IX_L \leftarrow (SP)$ $SP \rightarrow SP+2$	•	•	X	•	X	•	•	11 011 101 DD 11 100 001 E1	2	4	14	
POP IY	$IY_H \leftarrow (SP+1)$ $IY_L \leftarrow (SP)$ $SP \rightarrow SP+2$	•	•	X	•	X	•	•	11 111 101 FD 11 100 001 E1	2	4	14	

NOTE: (PAIR)<sub>H</sub>, (PAIR)<sub>L</sub> refer to high order and low order eight bits of the register pair respectively, e.g., BC<sub>L</sub> = C, AF<sub>H</sub> = A.

## EXCHANGE, BLOCK TRANSFER, BLOCK SEARCH GROUPS

Mnemonic	Symbolic Operation	S	Z	Flags	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments		
EX DE, HL	DE ↔ HL	•	•	X	•	X	•	•	11	101	011	EB	1	1	4		
EX AF, AF'	AF ↔ AF'	•	•	X	•	X	•	•	00	001	000	08	1	1	4		
EXX	BC ↔ BC' DE ↔ DE' HL ↔ HL'	•	•	X	•	X	•	•	11	011	001	D9	1	1	4	Register bank and auxiliary register bank exchange	
EX (SP), HL	H ↔ (SP+1) L ↔ (SP)	•	•	X	•	X	•	•	11	100	011	E3	1	5	19		
EX (SP), IX	IX <sub>H</sub> ↔ (SP+1) IX <sub>L</sub> ↔ (SP)	•	•	X	•	X	•	•	11	011	101	DD	2	6	23		
EX (SP), IY	IY <sub>H</sub> ↔ (SP+1) IY <sub>L</sub> ↔ (SP)	•	•	X	•	X	•	•	11	111	101	FD	2	6	23		
									11	100	011	E3					
LDI	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC - 1	•	•	X	0	X	①	0	•	11	101	101	ED	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
										10	100	000	A0				
LDIR	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC - 1 Repeat until BC = 0	•	•	X	0	X	②	0	•	11	101	101	ED	2	5	21	If BC ≠ 0
										10	110	000	B0	2	4	16	If BC = 0
LDD	(DE) ← (HL) DE ← DE - 1 HL ← HL - 1 BC ← BC - 1	•	•	X	0	X	①	0	•	11	101	101	ED	2	4	16	
										10	101	000	A8				
LDDR	(DE) ← (HL) DE ← DE - 1 HL ← HL - 1 BC ← BC - 1 Repeat until BC = 0	•	•	X	0	X	②	0	•	11	101	101	ED	2	5	21	If BC ≠ 0
										10	111	000	B8	2	4	16	If BC = 0
CPI	A - (HL) HL ← HL + 1 BC ← BC - 1	③	③	X	①	X	①	1	•	11	101	101	ED	2	4	16	
										10	100	001	A1				

NOTE: ① P/V flag is 0 if the result of BC - 1 = 0, otherwise P/V = 1.  
 ② P/V flag is 0 only at completion of instruction.  
 ③ Z flag is 1 if A = HL, otherwise Z = 0.

## EXCHANGE, BLOCK TRANSFER, BLOCK SEARCH GROUPS (Continued)

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments	
CPIR	A ← (HL)	⊡	⊡	X	⊡	X	⊡	1	•	11	101 101	ED	2	5	21	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)
	HL ← HL + 1									10	110 001	B1	2	4	16	
	BC ← BC - 1															
	Repeat until A = (HL) or BC = 0															
CPD	A ← (HL)	⊡	⊡	X	⊡	X	⊡	1	•	11	101 101	ED	2	4	16	
	HL ← HL - 1									10	101 001	A9				
	BC ← BC - 1															
CPDR	A ← (HL)	⊡	⊡	X	⊡	X	⊡	1	•	11	101 101	ED	2	5	21	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)
	HL ← HL - 1									10	111 001	B9	2	4	16	
	BC ← BC - 1															
	Repeat until A = (HL) or BC = 0															

NOTE: ① P/V flag is 0 if the result of BC - 1 = 0, otherwise P/V = 1.  
 ② P/V flag is 0 only at completion of instruction.  
 ③ Z flag is 1 if A = (HL), otherwise Z = 0.

## 8-BIT ARITHMETIC AND LOGICAL GROUP

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD A, r	A ← A + r	†	†	X	†	X	V	0	†	10	<u>000</u> r	1	1	4	r Reg.
ADD A, n	A ← A + n	†	†	X	†	X	V	0	†	11	<u>000</u> ← n →	2	2	7	000 B
															001 C
															010 D
															011 E
ADD A, (HL)	A ← A + (HL)	†	†	X	†	X	V	0	†	10	<u>000</u> 110	1	2	7	101 H
ADD A, (IX + d)	A ← A + (IX + d)	†	†	X	†	X	V	0	†	11	011 101 <u>000</u> ← d →	3	5	19	100 L
															101 A
															111 A
ADD A, (IY + d)	A ← A + (IY + d)	†	†	X	†	X	V	0	†	11	111 101 <u>000</u> ← d →	3	5	19	
ADC A, s	A ← A + s + CY	†	†	X	†	X	V	0	†		<u>001</u>				s is any of r, n, (HL), (IX + d), (IY + d) as shown for ADD instruction. The indicated bits replace the <u>000</u> in the ADD set above.
SUB s	A ← A - s	†	†	X	†	X	V	1	†		<u>010</u>				
SBC A, s	A ← A - s - CY	†	†	X	†	X	V	1	†		<u>011</u>				
AND s	A ← A > s	†	†	X	1	X	P	0	0		<u>100</u>				
OR s	A ← A > s	†	†	X	0	X	P	0	0		<u>110</u>				
XOR s	A ← A ⊕ s	†	†	X	0	X	P	0	0		<u>101</u>				
CP s	A - s	†	†	X	†	X	V	1	†		<u>111</u>				



## 8-BIT ARITHMETIC AND LOGICAL GROUP (Continued)

Mnemonic	Symbolic Operation	Flags			Opcode			Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments	
		S	Z	H	P/V	N	C						
INC r	$r \leftarrow r + 1$	†	†	X	†	X	V 0 •	00 r	100	1	1	4	
INC (HL)	(HL) $\leftarrow$ (HL) + 1	†	†	X	†	X	V 0 •	00 110	100	1	3	11	
INC (IX + d)	(IX + d) $\leftarrow$ (IX + d) + 1	†	†	X	†	X	V 0 •	11 011 101	DD	3	6	23	
								00 110	100				
								$\leftarrow d \rightarrow$					
INC (IY + d)	(IY + d) $\leftarrow$ (IY + d) + 1	†	†	X	†	X	V 0 •	11 111 101	FD	3	6	23	
								00 110	100				
								$\leftarrow d \rightarrow$					
DEC m	$m \leftarrow m - 1$	†	†	X	†	X	V 1 •		101				

NOTE: m is any of r, (HL), (IX+d), (IY+d) as shown for INC. DEC same format and states as INC. Replace 100 with 101 in opcode.

## GENERAL-PURPOSE ARITHMETIC AND CPU CONTROL GROUPS


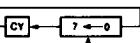
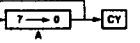
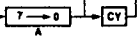
Mnemonic	Symbolic Operation	Flags			Opcode			Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments			
		S	Z	H	P/V	N	C						76	543	210
DAA	@	†	†	X	†	X	P • †	00	100	111	27	1	1	4	Decimal adjust accumulator
CPL	A ← A	•	•	X	1	X	• 1 •	00	101	111	2F	1	1	4	Complement accumulator (one's complement).
NEG	A ← 0 - A	†	†	X	†	X	V 1 †	11 01	101 000	101 100	ED 44	2	2	8	Negate acc. (two's complement).
CCF	CY ← CY	•	•	X	X	X	• 0 †	00	111	111	3F	1	1	4	Complement carry flag.
SCF	CY ← 1	•	•	X	0	X	• 0 1	00	110	111	37	1	1	4	Set carry flag.
NOP	No operation	•	•	X	•	X	• • •	00	000	000	00	1	1	4	
HALT	CPU halted	•	•	X	•	X	• • •	01	110	110	76	1	1	4	
DI ★	IFF ← 0	•	•	X	•	X	• • •	11	110	011	F3	1	1	4	
EI ★	IFF ← 1	•	•	X	•	X	• • •	11	111	011	FB	1	1	4	
IM 0	Set interrupt mode 0	•	•	X	•	X	• • •	11 01	101 000	101 110	ED 46	2	2	8	
IM 1	Set interrupt mode 1	•	•	X	•	X	• • •	11	101	101	ED	2	2	8	
								01	010	110	56				
IM 2	Set interrupt mode 2	•	•	X	•	X	• • •	11 01	101 011	101 110	ED 5E	2	2	8	

NOTES: @ converts accumulator content into packed BCD following add or subtract with packed BCD operands.  
 IFF indicates the interrupt enable flip-flop.  
 CY indicates the carry flip-flop.  
 \* indicates interrupts are not sampled at the end of EI or DI.

## 16-BIT ARITHMETIC GROUP

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD HL, ss	HL ← HL + ss	•	•	X	X	X	•	0 ‡	00 ssl 001	1	3	11	ss Reg. 00 BC
ADC HL, ss	HL ← HL + ss + CY	‡	‡	X	X	X	V	0 ‡	11 101 101 ED 01 ss1 010	2	4	15	01 DE 10 HL 11 SP
SBC HL, ss	HL ← HL - ss - CY	‡	‡	X	X	X	V	1 ‡	11 101 101 ED 01 ss0 010	2	4	15	
ADD IX, pp	IX ← IX + pp	•	•	X	X	X	•	0 ‡	11 011 101 DD 01 pp1 001	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY, rr	IY ← IY + rr	•	•	X	X	X	•	0 ‡	11 111 101 FD 00 rr1 001	2	4	15	rr Reg. 00 BC
INC ss	ss ← ss + 1	•	•	X	•	X	•	•	00 ss0 011	1	1	6	01 DE
INC IX	IX ← IX + 1	•	•	X	•	X	•	•	11 011 101 DD 00 100 011 23	2	2	10	10 IY 11 SP
INC IY	IY ← IY + 1	•	•	X	•	X	•	•	11 111 101 FD 00 100 011 23	2	2	10	
DEC ss	ss ← ss - 1	•	•	X	•	X	•	•	00 ss1 011	1	1	6	
DEC IX	IX ← IX - 1	•	•	X	•	X	•	•	11 011 101 DD 00 101 011 2B	2	2	10	
DEC IY	IY ← IY - 1	•	•	X	•	X	•	•	11 111 101 FD 00 101 011 2B	2	2	10	

## ROTATE AND SHIFT GROUP

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLCA		•	•	X	0	X	•	0 ‡	00 000 111 07	1	1	4	Rotate left circular accumulator.
RLA		•	•	X	0	X	•	0 ‡	00 010 111 17	1	1	4	Rotate left accumulator.
RRCA		•	•	X	0	X	•	0 ‡	00 001 111 0F	1	1	4	Rotate right circular accumulator.
RRA		•	•	X	0	X	•	0 ‡	00 011 111 1F	1	1	4	Rotate right accumulator.

# ROTATE AND SHIFT GROUP (Continued)

Symbolic Mnemonic Operation	S	Z	Flags H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLC r			X	0	X	P 0	11	001	011	CB	2	2	8	Rotate left circular register r.
							00	000	r					
RLC (HL)			X	0	X	P 0	11	001	011	CB	2	4	15	r Reg.
							00	000	110					000 B
RLC (IX+d)			X	0	X	P 0	11	011	101	DD	4	6	23	001 C
							11	001	011	CB				010 D
														011 E
														001 H
														101 L
														111 A
RLC (IY+d)			X	0	X	P 0	11	111	101	FD	4	6	23	
							11	001	011	CB				
RL m			X	0	X	P 0	11	001	011	CB				
							00	000	110					
												</		

## BIT SET, RESET AND TEST GROUP

Mnemonic	Symbolic Operation	S	Z	Flags				Opcode			Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
				H	P/V	N	C	76	543	210					
BIT b, r	$Z \leftarrow r_b$	X	†	X	1	X	X	0	•	11 001 011	CB	2	2	8	r Reg.
								01	b	r					000 B
BIT b, (HL)	$Z \leftarrow (HL)_b$	X	†	X	1	X	X	0	•	11 001 011	CB	2	3	12	001 C
								01	b	110					010 D
BIT b, (IX+d) <sub>b</sub>	$Z \leftarrow (IX+d)_b$	X	†	X	1	X	X	0	•	11 011 101	DD	4	5	20	011 E
								11	001 011	CB					100 H
								$\leftarrow d \rightarrow$							101 L
								01	b	110					111 A
								b Bit Tested							
BIT b, (IY+d) <sub>b</sub>	$Z \leftarrow (IY+d)_b$	X	†	X	1	X	X	0	•	11 111 101	FD	4	5	20	000 0
								11	001 011	CB					001 1
								$\leftarrow d \rightarrow$							010 2
								01	b	110					011 3
SET b, r	$r_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 001 011	CB	2	2	8	100 4
								11	b	r					101 5
SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 001 011	CB	2	4	15	110 6
								11	b	110					111 7
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 011 101	DD	4	6	23	
								11	001 011	CB					
								$\leftarrow d \rightarrow$							
								11	b	110					
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 111 101	FD	4	6	23	
								11	001 011	CB					
								$\leftarrow d \rightarrow$							
								11	b	110					
RES b, m	$m_b \leftarrow 0$	•	•	X	•	X	•	•	•	11 101 101	FD	4	6	23	
	$m = r, (HL),$							11	b	110					
	$(IX+d), (IY+d)$							10							

To form new opcode replace 11 of SET b, s with 10. Flags and time states for SET instruction.

To form new opcode replace **11** of SET b, s with **10**. Flags and time states for SET instruction.

NOTE: The notation  $m_b$  indicates location m, bit b (0 to 7).

## JUMP GROUP

Mnemonic	Symbolic Operation	S	Z	Flags			Opcode				Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments		
				H	P/V/N	C	76	543	210								
JP nn	PC ← nn	•	•	X	•	X	•	•	•	•	11 000 011	C3	3	3	10	cc Condition 000 NZ (non-zero) 001 Z (zero)	
JP cc, nn	If condition cc is true PC←nn, otherwise continue	•	•	X	•	X	•	•	•	•	11 cc 010		3	3	10	010 NC (non-carry) 011 C (carry) 100 PO (parity odd) 101 PE (parity even)	
											←n→						110 P (sign positive)
											←n→						111 M (sign negative)
											←n→						
											←n→						
JR e	PC ← PC+e	•	•	X	•	X	•	•	•	•	00 011 000	18	2	3	12	110 P (sign positive) 111 M (sign negative)	
JR C, e	If C=0, continue If C=1, PC ← PC+e	•	•	X	•	X	•	•	•	•	00 111 000	38	2	2	7	If condition not met.	
											←e-2→						
												2	3	12	If condition is met.		
JR NC, e	If C=1, continue If C=0, PC ← PC+e	•	•	X	•	X	•	•	•	•	00 110 000	30	2	2	7	If condition not met.	
											←e-2→						
												2	3	12	If condition is met.		
JP Z, e	If Z=0, continue If Z=1, PC ← PC+e	•	•	X	•	X	•	•	•	•	00 101 000	28	2	2	7	If condition not met.	
											←e-2→						
												2	3	12	If condition is met.		
JR NZ, e	If Z=1, continue If Z=0, PC ← PC+e	•	•	X	•	X	•	•	•	•	00 100 000	20	2	2	7	If condition not met.	
											←e-2→						
												2	3	12	If condition is met.		
JP (HL)	PC ← HL	•	•	X	•	X	•	•	•	•	11 101 001	E9	1	1	4		
JP (IX)	PC ← IX	•	•	X	•	X	•	•	•	•	11 011 101	DD	2	2	8		
											11 101 001	E9					
JP (IY)	PC ← IY	•	•	X	•	X	•	•	•	•	11 111 101	FD	2	2	8		
											11 101 001	E9					
DJNZ, e	B ← B-1 If B=0, continue If B≠0, PC ← PC+e	•	•	X	•	X	•	•	•	•	00 010 000	10	2	2	8	If B=0	
											←e-2→						
												2	3	13	If B≠0.		

NOTES: e represents the extension in the relative addressing mode.

e is a signed two's complement number in the range < -128, 127 >.

e - 2 in the opcode provides an effective address of pc + e as PC is incremented by 2 prior to the addition of e.

**CALL AND RETURN GROUP**

[illegible]

NOTE: <sup>1</sup>RETN loads IFF<sub>2</sub> → IFF<sub>1</sub>

## INPUT AND OUTPUT GROUP

Mnemonic	Symbolic Operation	Flags					Opcode				No. of Bytes	No. of M Cycles	No. of T States	Comments				
		S	Z	H	P/VN	C	76	543	210	Hex								
IN A, (n)	A ← (n)	•	•	X	•	X	•	•	•	11	011	01	DB	2	3	11	n to A <sub>0</sub> ~ A <sub>7</sub> Acc. to A <sub>8</sub> ~ A <sub>15</sub>	
IN r, (C)	r ← (C) if r = 110 only the flags will be affected	†	†	X	†	X	P	0	•	11	101	101	ED	2	3	12	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>	
										01	r	000						
INI	(HL) ← (C) B ← B - 1 HL ← HL + 1	①	X	†	X	X	X	X	1	X	11	101	101	ED	2	4	16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
											10	100	010	A2				
INIR	(HL) ← (C) B ← B - 1 HL ← HL + 1 Repeat until B = 0	②	X	1	X	X	X	X	1	X	11	101	101	ED	2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
											10	110	010	B2				
IND	(HL) ← (C) B ← B - 1 HL ← HL - 1	①	X	†	X	X	X	X	1	X	11	101	101	ED	2	4	16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
											10	101	010	AA				
INDR	(HL) ← (C) B ← B - 1 HL ← HL - 1 Repeat until B = 0	②	X	1	X	X	X	X	1	X	11	101	101	ED	2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
											10	111	010	BA				
OUT (n), A	(n) → A		•	•	X	•	X	•	•	•	11	010	011	D3	2	3	11	n to A <sub>0</sub> ~ A <sub>7</sub> Acc. to A <sub>8</sub> ~ A <sub>15</sub>
OUT (C), r	(C) → r		•	•	X	•	X	•	•	•	11	101	101	ED	2	3	12	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
											01	r	001					
OUTI	(C) → (HL) B ← B - 1 HL ← HL + 1	①	X	†	X	X	X	X	1	X	11	101	101	ED	2	4	16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
											10	100	011	A3				
OTIR	(C) → (HL) B ← B - 1 HL ← HL + 1 Repeat until B = 0	②	X	1	X	X	X	X	1	X	11	101	101	ED	2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
											10	110	011	B3				
OUTD	(C) → (HL) B ← B - 1 HL ← HL - 1	①	X	†	X	X	X	X	1	X	11	101	101	ED	2	4	16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
											10	101	011	AB				
OTDR	(C) → (HL) B ← B - 1 HL ← HL - 1 Repeat until B = 0	②	X	1	X	X	X	X	1	X	11	101	101	ED	2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
											10	111	011					

NOTES: ① If the result of B - 1 is zero, the Z flag is set; otherwise it is reset.  
② Z flag is set upon instruction completion only.

## SUMMARY OF FLAG OPERATION

Instructions	D <sub>7</sub> S	Z	H	P/V	N	D <sub>0</sub> C	Comments
ADD A, s; ADC A, s	†	†	X	†	X	V 0	8-bit add or add with carry.
SUB s; SBC A, s; CP s; NEG	†	†	X	†	X	V 1	8-bit subtract, subtract with carry, compare and negate accumulator.
AND s	†	†	X	1	X	P 0 0	Logical operation.
OR s, XOR s	†	†	X	0	X	P 0 0	Logical operation.
INC s	†	†	X	†	X	V 0 •	8-bit increment.
DEC s	†	†	X	†	X	V 1 •	8-bit decrement.
ADD DD, ss	•	•	X	X	X	• 0 †	16-bit add.
ADC HL, ss	†	†	X	X	X	V 0 †	16-bit add with carry.
SBC HL, ss	†	†	X	X	X	V 1 †	16-bit subtract with carry.
RLA; RLCA; RRA; RRCA	•	•	X	0	X	• 0 †	Rotate accumulator.
RL m; RLC m; RR m; RRC m; SLA m; SRA m; SRL m	†	†	X	0	X	P 0 †	Rotate and shift locations.
RLD; RRD	†	†	X	0	X	P 0 •	Rotate digit left and right.
DAA	†	†	X	†	X	P • †	Decimal adjust accumulator.
CPL	•	•	X	1	X	• 1 •	Complement accumulator.
SCF	•	•	X	0	X	• 0 1	Set carry.
CCF	•	•	X	X	X	• 0 †	Complement carry.
IN r (C)	†	†	X	0	X	P 0 •	Input register indirect.
INI; IND; OUTI; OUTD	X	†	X	X	X	X 1 •	Block input and output. Z = 1 if B ≠ 0, otherwise Z = 0.
INIR; INDR; OTIR; OTDR	X	1	X	X	X	X 1 •	Block input and output. Z = 1 if B ≠ 0, otherwise Z = 0.
LDI; LDD	X	X	X	0	X	† 0 •	Block transfer instructions. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
LDIR; LDDR	X	X	X	0	X	0 0 •	Block transfer instructions. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
CPI; CPIR; CPD; CPDR	X	†	X	X	X	† 1 •	Block search instructions. Z = 1 if A = (HL), otherwise Z = 0. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
LD A, I; LD A, R	†	†	X	0	X	IFF 0 •	IFF, the content of the interrupt enable flip-flop, (IFF <sub>2</sub> ), is copied into the P/V flag.
BIT b, s	X	†	X	1	X	X 0 •	The state of bit b of location s is copied into the Z flag.

## SYMBOLIC NOTATION

Symbol	Operation	Symbol	Operation
S	Sign flag. S = 1 if the MSB of the result is 1.	†	The flag is affected according to the result of the operation.
Z	Zero flag. Z = 1 if the result of the operation is 0.	•	The flag is unchanged by the operation.
P/V	Parity or overflow flag. Parity (P) and overflow (V) share the same flag. Logical operations affect this flag with the parity of the result while arithmetic operations affect this flag with the overflow of the result. If P/V holds parity: P/V = 1 if the result of the operation is even; P/V = 0 if result is odd. If P/V holds overflow, P/V = 1 if the result of the operation produced an overflow. If P/V does not hold overflow, P/V = 0.	0	The flag is reset by the operation.
H*	Half-carry flag. H = 1 if the add or subtract operation produced a carry into, or borrow from, bit 4 of the accumulator.	1	The flag is set by the operation.
N*	Add/Subtract flag. N = 1 if the previous operation was a subtract.	X	The flag is indeterminate.
C	Carry/Link flag. C = 1 if the operation produced a carry from the MSB of the operand or result.	V	P/V flag affected according to the overflow result of the operation.
		P	P/V flag affected according to the parity result of the operation.
		r	Any one of the CPU registers A, B, C, D, E, H, L.
		s	Any 8-bit location for all the addressing modes allowed for the particular instruction.
		ss	Any 16-bit location for all the addressing modes allowed for that instruction.
		ii	Any one of the two index registers IX or IY.
		R	Refresh counter.
		n	8-bit value in range < 0, 255 >.
		nn	16-bit value in range < 0, 65535 >.

\* H and N flags are used in conjunction with the decimal adjust instruction (DAA) to properly correct the result into packed BCD format following addition or subtraction using operands with packed BCD format.



## CPU REGISTERS

Figure 4 shows three groups of registers within the CPU. The first group consists of duplicate sets of 8-bit registers: a principal set and an alternate set [designated by ' (prime), e.g., A']. Both sets consist of the Accumulator register, the Flag register, and six general-purpose registers. Transfer of data between these duplicate sets of registers is accomplished by use of "Exchange" instructions. The result is faster response to interrupts and easy, efficient implementation of such versatile programming techniques

as background-foreground data processing. The second set of registers consists of six registers with assigned functions. These are the I (Interrupt register), the R (Refresh register), the IX and IY (Index registers), the SP (Stack Pointer), and the PC (Program Counter). The third group consists of two interrupt status flip-flops, plus an additional pair of flip-flops which assists in identifying the interrupt mode at any particular time. Table 1 provides further information on these registers.

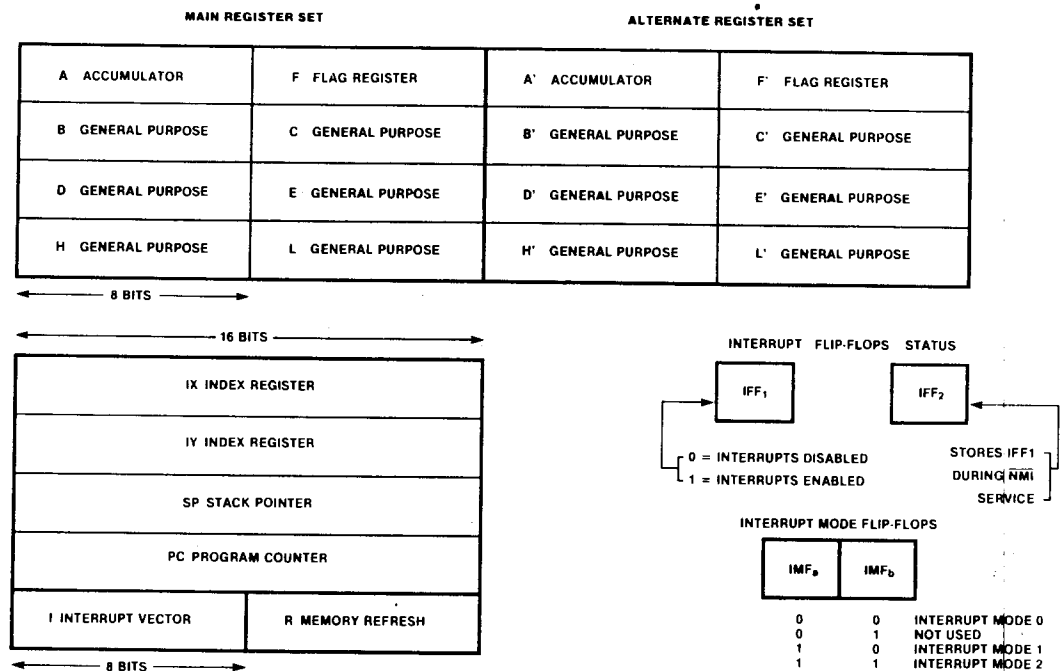


Figure 4. CPU Registers

## INTERRUPTS: GENERAL OPERATION

The CPU accepts two interrupt input signals:  $\overline{\text{NMI}}$  and  $\overline{\text{INT}}$ . The  $\overline{\text{NMI}}$  is a non-maskable interrupt and has the highest priority.  $\overline{\text{INT}}$  is a lower priority interrupt and it requires that interrupts be enabled in software in order to operate.  $\overline{\text{INT}}$  can be connected to multiple peripheral devices in a wired-OR configuration.

The Z80 has a single response mode for interrupt service on the non-maskable interrupt. The maskable interrupt,  $\overline{\text{INT}}$ , has three programmable response modes available. These are:

- Mode 0 — similar to the 8080 microprocessor.
- Mode 1 — Peripheral Interrupt service, for use with non-8080/Z80 systems.

- Mode 2 - a vectored interrupt scheme, usually daisy-chained, for use with the Z80 Family and compatible peripheral devices.

The CPU services interrupts by sampling the  $\overline{\text{NMI}}$  and  $\overline{\text{INT}}$  signals at the rising edge of the last clock of an instruction. Further interrupt service processing depends upon the type of interrupt that was detected. Details on interrupt responses are shown in the CPU Timing Section.

**Non-Maskable Interrupt ( $\overline{\text{NMI}}$ ).** The nonmaskable interrupt cannot be disabled by program control and therefore will be accepted at all times by the CPU.  $\overline{\text{NMI}}$  is usually reserved for servicing only the highest priority type interrupts, such as that for orderly shutdown after power

---

## PIN DESCRIPTIONS

**A<sub>0</sub>-A<sub>15</sub>.** *Address Bus* (output, active High, 3-state). A<sub>0</sub>-A<sub>15</sub> form a 16-bit address bus. The Address Bus provides the address for memory data bus exchanges (up to 64K bytes) and for I/O device exchanges.

**BUSACK.** *Bus Acknowledge* (output, active Low). Bus Acknowledge indicates to the requesting device that the CPU address bus, data bus, and control signals  $\overline{\text{MREQ}}$ ,  $\overline{\text{IORQ}}$ ,  $\overline{\text{RD}}$ , and  $\overline{\text{WR}}$  have entered their high-impedance states. The external circuitry can now control these lines.

**BUSREQ.** *Bus Request* (input, active Low). Bus Request has a higher priority than  $\overline{\text{NMI}}$  and is always recognized at the end of the current machine cycle.  $\overline{\text{BUSREQ}}$  forces the CPU address bus, data bus, and control signals  $\overline{\text{MREQ}}$ ,  $\overline{\text{IORQ}}$ ,  $\overline{\text{RD}}$ , and  $\overline{\text{WR}}$  to go to a high-impedance state so that other devices can control these lines.  $\overline{\text{BUSREQ}}$  is normally wired-OR and requires an external pullup for these applications. Extended  $\overline{\text{BUSREQ}}$  periods due to extensive DMA operations can prevent the CPU from properly refreshing dynamic RAMs.

**D<sub>0</sub>-D<sub>7</sub>.** *Data Bus* (input/output, active High, 3-state). D<sub>0</sub>-D<sub>7</sub> constitute an 8-bit bidirectional data bus, used for data exchanges with memory and I/O.

**HALT.** *Halt State* (output, active Low).  $\overline{\text{HALT}}$  indicates that the CPU has executed a Halt instruction and is awaiting either a nonmaskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOPs to maintain memory refresh.

**INT.** *Interrupt Request* (input, active Low). Interrupt Request is generated by I/O devices. The CPU honors a request at the end of the current instruction if the internal software-controlled interrupt enable flip-flop (IFF) is enabled.  $\overline{\text{INT}}$  is normally wired-OR and requires an external pullup for these applications.

**IORQ.** *Input/Output Request* (output, active Low, 3-state).  $\overline{\text{IORQ}}$  indicates that the lower half of the address bus holds a valid I/O address for an I/O read or write operation.  $\overline{\text{IORQ}}$  is also generated concurrently with  $\overline{\text{M1}}$  during an interrupt acknowledge cycle to indicate that an interrupt response vector can be placed on the data bus.

**M1.** *Machine Cycle One* (output, active Low).  $\overline{\text{M1}}$ , together with  $\overline{\text{MREQ}}$ , indicates that the current machine cycle is the opcode fetch cycle of an instruction execution.  $\overline{\text{M1}}$ , together with  $\overline{\text{IORQ}}$ , indicates an interrupt acknowledge cycle.

**MREQ.** *Memory Request* (output, active Low, 3-state).  $\overline{\text{MREQ}}$  indicates that the address bus holds a valid address for a memory read or memory write operation.

**NMI.** *Non-Maskable Interrupt* (input, negative edge-triggered).  $\overline{\text{NMI}}$  has a higher priority than  $\overline{\text{INT}}$ .  $\overline{\text{NMI}}$  is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop, and automatically forces the CPU to restart at location 0066H.

**RD.** *Read* (output, active Low, 3-state).  $\overline{\text{RD}}$  indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.

**RESET.** *Reset* (input, active Low).  $\overline{\text{RESET}}$  initializes the CPU as follows: it resets the interrupt enable flip-flop, clears the PC and Registers I and R, and sets the interrupt status to Mode 0. During reset time, the address and data bus go to a high-impedance state, and all control output signals go to the inactive state. Note that  $\overline{\text{RESET}}$  must be active for a minimum of three full clock cycles before the reset operation is complete.

**RFSH.** *Refresh* (output, active Low).  $\overline{\text{RFSH}}$ , together with  $\overline{\text{MREQ}}$ , indicates that the lower seven bits of the system's address bus can be used as a refresh address to the system's dynamic memories.

**WAIT.** *Wait* (input, active Low).  $\overline{\text{WAIT}}$  indicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter a Wait state as long as this signal is active. Extended  $\overline{\text{WAIT}}$  periods can prevent the CPU from properly refreshing dynamic memory.

**WR.** *Write* (output, active Low, 3-state).  $\overline{\text{WR}}$  indicates that the CPU data bus holds valid data to be stored at the addressed memory or I/O location.

## CPU TIMING

The Z80 CPU executes instructions by proceeding through a specific sequence of operations:

- Memory read or write
- I/O device read or write
- Interrupt acknowledge

The basic clock period is referred to as a T time or cycle, and three or more T cycles make up a machine cycle (M1, M2 or M3 for instance). Machine cycles can be extended either by the CPU automatically inserting one or more Wait states or by the insertion of one or more Wait states by the user.

**Instruction Opcode Fetch.** The CPU places the contents of the Program Counter (PC) on the address bus at the start of the cycle (Figure 5). Approximately one-half clock cycle later,  $\overline{\text{MREQ}}$  goes active. When active,  $\overline{\text{RD}}$  indicates that the memory data can be enabled onto the CPU data bus.

The CPU samples the  $\overline{\text{WAIT}}$  input with the falling edge of clock state T<sub>2</sub>. During clock states T<sub>3</sub> and T<sub>4</sub> of an M1 cycle, dynamic RAM refresh can occur while the CPU starts decoding and executing the instruction. When the Refresh Control signal becomes active, refreshing of dynamic memory can take place.

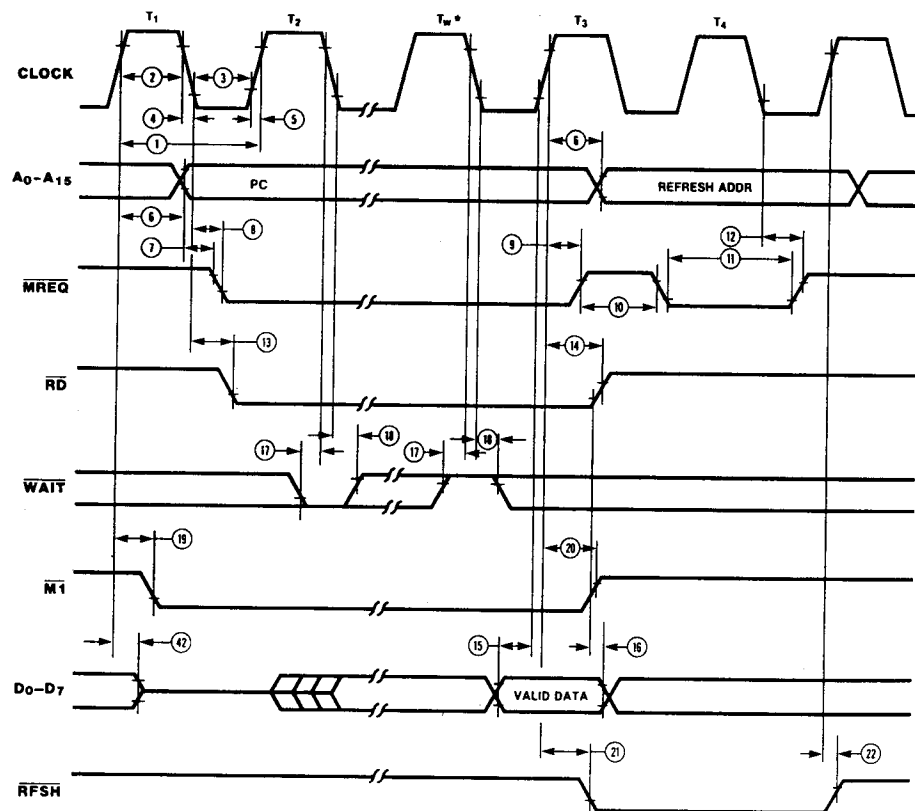


Figure 5. Instruction Opcode Fetch

**Memory Read or Write Cycles.** Figure 6 shows the timing of memory read or write cycles other than an opcode fetch (M1) cycle. The  $\overline{\text{MREQ}}$  and  $\overline{\text{RD}}$  signals function exactly as in the fetch cycle. In a memory write cycle,  $\overline{\text{MREQ}}$  also

becomes active when the address bus is stable. The  $\overline{\text{WR}}$  line is active when the data bus is stable, so that it can be used directly as an R/W pulse to most semiconductor memories.

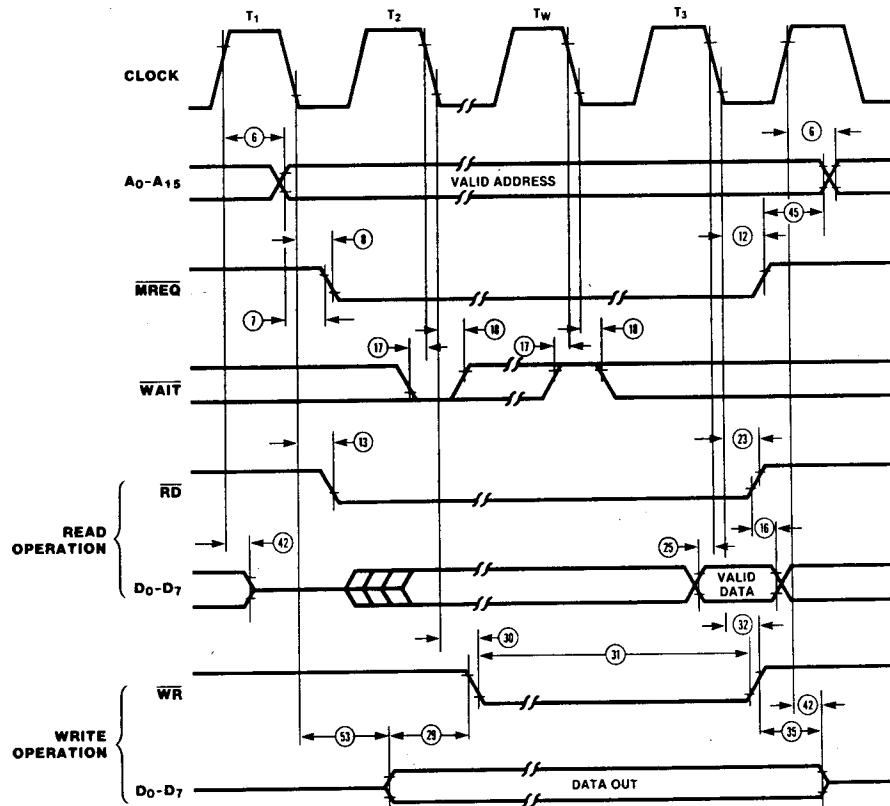
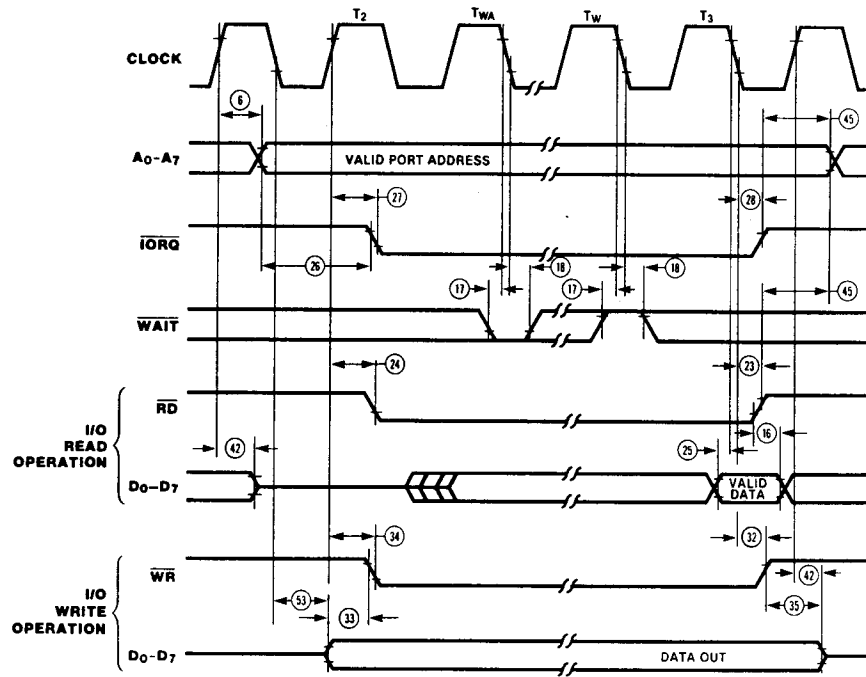


Figure 6. Memory Read or Write Cycles

**Input or Output Cycles.** Figure 7 shows the timing for an I/O read or I/O write operation. During I/O operations, the CPU automatically inserts a single Wait state ( $T_{WA}$ ). This

extra Wait state allows sufficient time for an I/O port to decode the address from the port address lines.

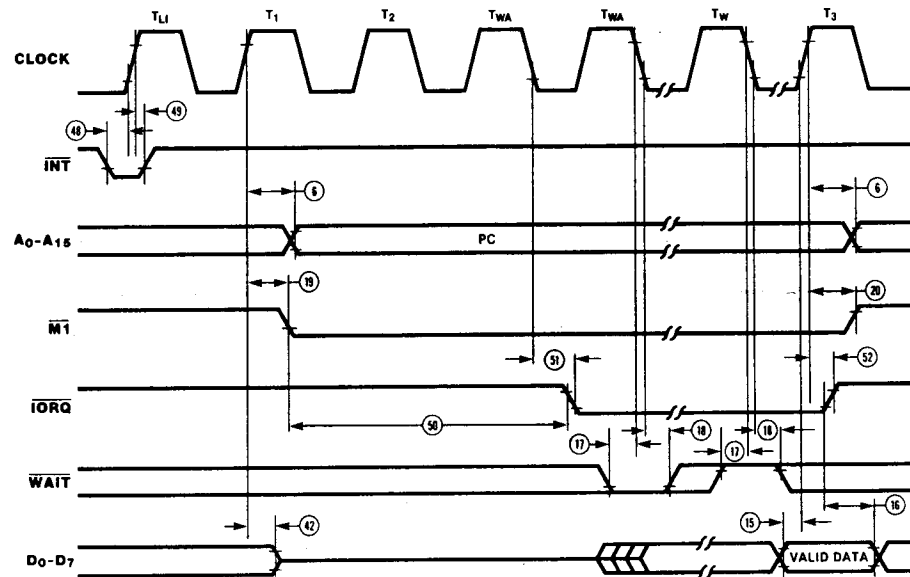


$T_{WA}$  = One wait cycle automatically inserted by CPU.

**Figure 7. Input or Output Cycles**

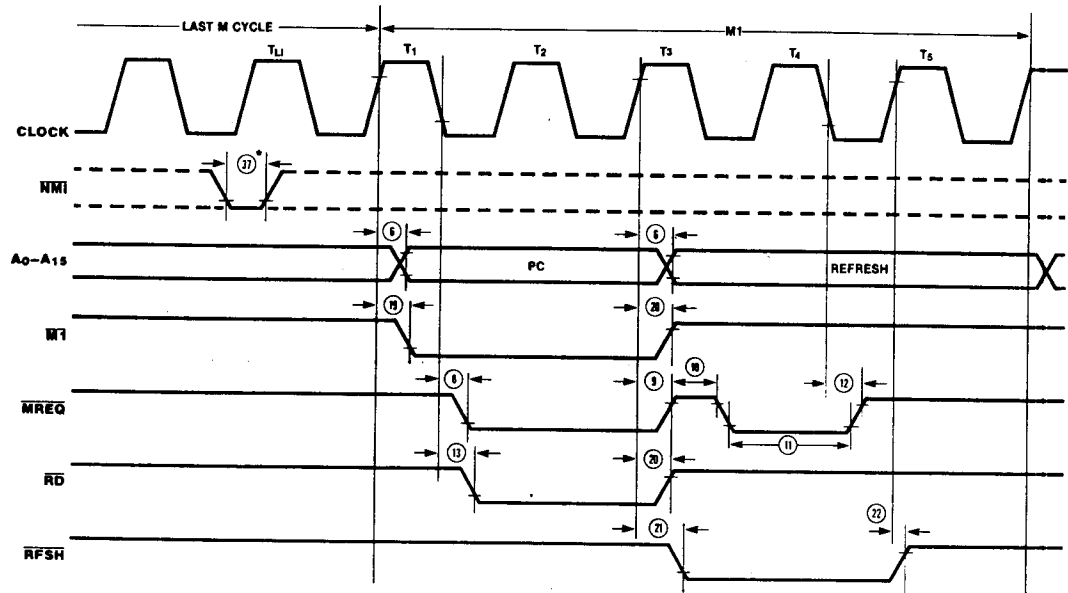
**Interrupt Request/Acknowledge Cycle.** The CPU samples the interrupt signal with the rising edge of the last clock cycle at the end of any instruction (Figure 8). When an interrupt is accepted, a special  $\overline{M1}$  cycle is generated.

During this  $\overline{M1}$  cycle,  $\overline{IORQ}$  becomes active (instead of  $\overline{MREQ}$ ) to indicate that the interrupting device can place an 8-bit vector on the data bus. The CPU automatically adds two Wait states to this cycle.



**Non-Maskable Interrupt Request Cycle.**  $\overline{\text{NMI}}$  is sampled at the same time as the maskable interrupt input  $\overline{\text{INT}}$  but has higher priority and cannot be disabled under software control. The subsequent timing is similar to that of a normal

memory read operation except that data put on the bus by the memory is ignored. The CPU instead executes a restart (RST) operation and jumps to the  $\overline{\text{NMI}}$  service routine located at address 0066H (Figure 9).

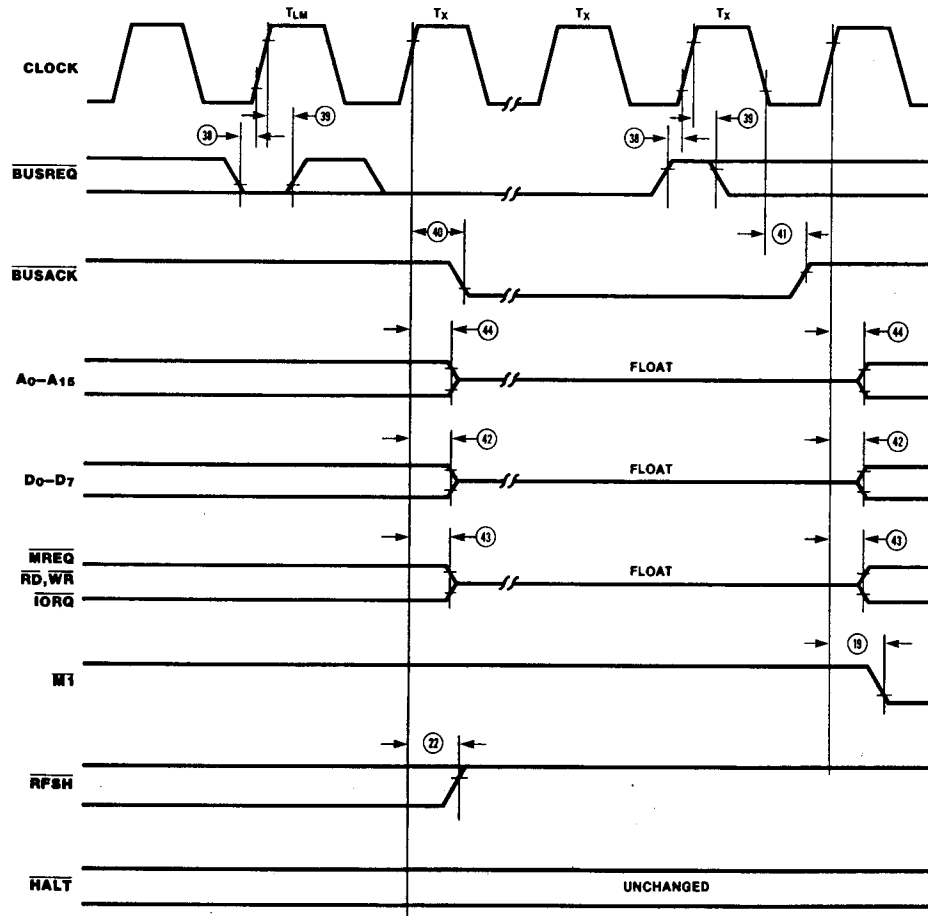


\*Although  $\overline{\text{NMI}}$  is an asynchronous input, to guarantee its being recognized on the following machine cycle,  $\overline{\text{NMI}}$ 's falling edge must occur no later than the rising edge of the clock cycle preceding the last state of any instruction cycle (T<sub>LJ</sub>).

Figure 9. Non-Maskable Interrupt Request Operation

**Bus Request/Acknowledge Cycle.** The CPU samples  $\overline{\text{BUSREQ}}$  with the rising edge of the last clock period of any machine cycle (Figure 10). If  $\overline{\text{BUSREQ}}$  is active, the CPU sets its address, data, and  $\overline{\text{MREQ}}$ ,  $\overline{\text{IORQ}}$ , RD, and WR lines

to a high-impedance state with the rising edge of the next clock pulse. At that time, any external device can take control of these lines, usually to transfer data between memory and I/O devices.



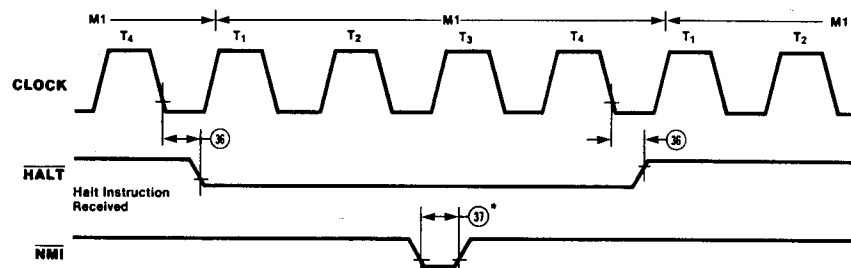
NOTES: 1)  $T_{LM}$  = Last state of any M cycle.  
2)  $T_X$  = An arbitrary clock cycle used by requesting device.

Figure 10. BUS Request/Acknowledge Cycle



**Halt Acknowledge Cycle.** When the CPU receives a HALT instruction, it executes NOP states until either an INT or NMI input is received. When in the Halt state, the HALT output is

active and remains so until an interrupt is received (Figure 11). INT will also force a Halt exit.



\*Although NMI is an asynchronous input, to guarantee its being recognized on the following machine cycle, NMI's falling edge must occur no later than the rising edge of the clock cycle preceding the last state of any instruction cycle ( $T_{LJ}$ ).

Figure 11. Halt Acknowledge

**Reset Cycle.** RESET must be active for at least three clock cycles for the CPU to properly accept it. As long as RESET remains active, the address and data buses float, and the control outputs are inactive. Once RESET goes inactive, two

internal T cycles are consumed before the CPU resumes normal processing operation. RESET clears the PC register, so the first opcode fetch will be to location 0000H (Figure 12).

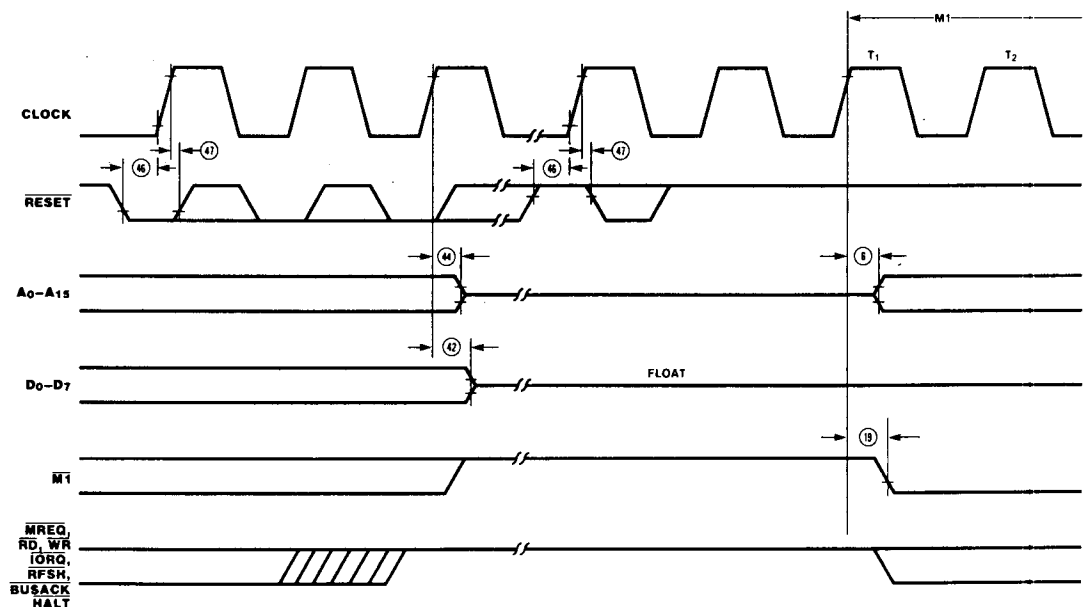


Figure 12. Reset Cycle

**Power-Down mode of operation** (Only applies to CMOS Z80 CPU).

CMOS Z80 CPU supports Power-Down mode of operation.

This mode is also referred to as the "standby mode", and supply current for the CPU goes down as low as 10  $\mu\text{A}$  (Where specified as  $I_{CC2}$ ).

**Power-Down Acknowledge Cycle.** When the clock input to the CPU is stopped at either a High or Low level, the CPU stops its operation and maintains all registers and control signals. However,  $I_{CC2}$  (standby supply current) is guaranteed only when the system clock is stopped at a Low

level during  $T_4$  of the machine cycle following the execution of the HALT instruction. The timing diagram for the power-down function, when implemented with the HALT instruction, is shown in Figure 13.

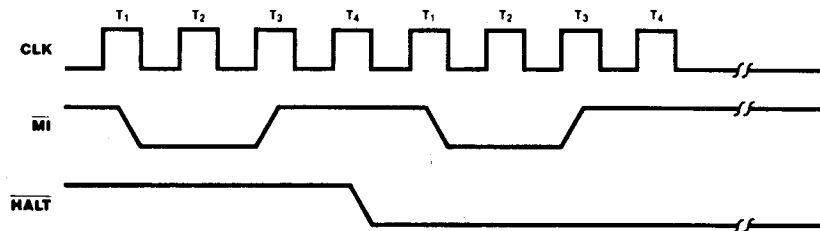


Figure 13. Power-Down Acknowledge

**Power-Down Release Cycle.** The system clock must be supplied to the CPU to release the power-down state. When the system clock is supplied to the CLK input, the CPU restarts operations from the point at which the power-down state was implemented. The timing diagrams for the release from power-down mode are shown in Figure 14.

**NOTES:**

- 1) When the external oscillator has been stopped to enter the power-down state, some warm-up time may be required to obtain a stable clock for the release.
- 2) When the HALT instruction is executed to enter the power-down state, the CPU will also enter the Halt state. An interrupt signal (either  $\overline{\text{NMI}}$  or  $\overline{\text{INT}}$ ) or a RESET signal must be applied to the CPU after the system clock is supplied in order to release the power-down state.

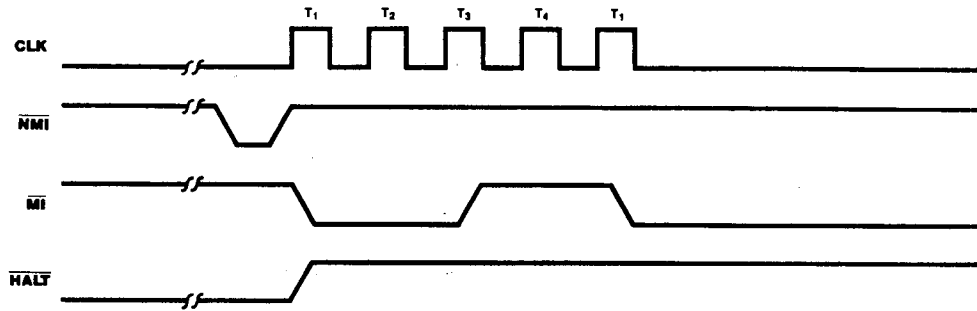


Figure 14a.

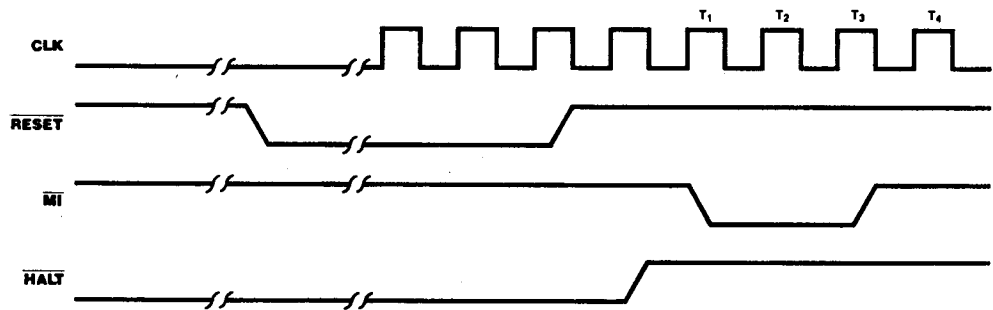


Figure 14b.

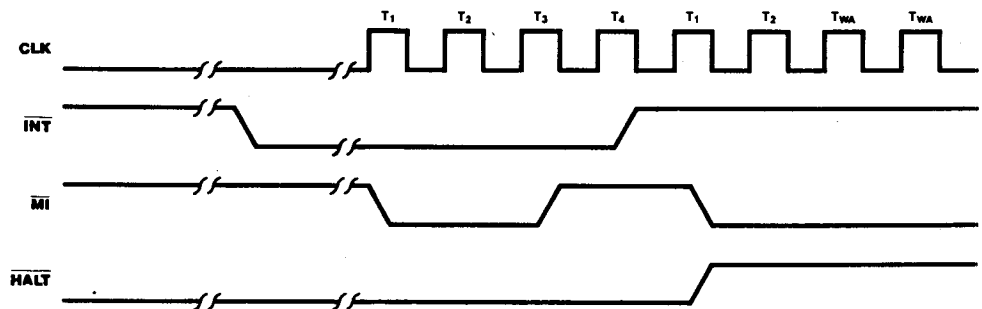


Figure 14c.

Figure 13. Power-Down Release

---

## ABSOLUTE MAXIMUM RATINGS

Voltage on  $V_{CC}$  with respect to  $V_{SS}$  . . . . .  $-0.3V$  to  $+7V$   
Voltages on all inputs with respect  
to  $V_{SS}$  . . . . .  $-0.3V$  to  $V_{CC} + 0.3V$   
Operating Ambient  
Temperature . . . . . See Ordering Information  
Storage Temperature . . . . .  $-65^{\circ}C$  to  $+150^{\circ}C$

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

---

## STANDARD TEST CONDITIONS

The DC Characteristics and capacitance sections below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0V). Positive current flows into the referenced pin.

The Ordering Information section lists temperature ranges and product numbers. Package drawings are in the Package Information section. Refer to the Literature List for additional documentation.

Available operating temperature ranges are:

■ **S =  $0^{\circ}C$  to  $+70^{\circ}C$**

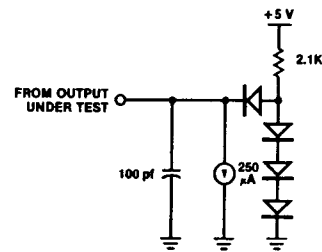
**Voltage Supply Range:**

NMOS:  $+4.75V \leq V_{CC} \leq +5.25V$

CMOS:  $+4.50V \leq V_{CC} \leq +5.50V$

■ **E =  $-40^{\circ}C$  to  $100^{\circ}C$ ,  $+4.50V \leq V_{CC} \leq +5.50V$**

All ac parameters assume a load capacitance of 100 pf. Add 10 ns delay for each 50 pf increase in load up to a maximum of 200 pf for the data bus and 100 pf for address and control lines. AC timing measurements are referenced to 1.5 volts (except for clock, which is referenced to the 10% and 90% points).



## DC CHARACTERISTICS (Z84C00/CMOS Z80 CPU)

Symbol	Parameter	Min	Max	Unit	Condition
$V_{ILC}$	Clock Input Low Voltage	-0.3	0.45	V	
$V_{IHC}$	Clock Input High Voltage	$V_{CC} - .6$	$V_{CC} + .3$	V	
$V_{IL}$	Input Low Voltage	-0.3	0.8	V	
$V_{IH}$	Input High Voltage	2.2	$V_{CC}$	V	
$V_{OL}$	Output Low Voltage		0.4	V	$I_{OL} = 2.0 \text{ mA}$
$V_{OH1}$	Output High Voltage	2.4		V	$I_{OH} = -1.6 \text{ mA}$
$V_{OH2}$	Output High Voltage	$V_{CC} - 0.8$		V	$I_{OH} = -250 \mu\text{A}$
$I_{CC1}$	Power Supply Current	4 MHz	20	mA	$V_{CC} = 5\text{V}$
			30	mA	$V_{IH} = V_{CC} - 0.2\text{V}$
			40	mA	$V_{IL} = 0.2\text{V}$
			50	mA	$V_{CC} = 5\text{V}$
			100	mA	$V_{CC} = 5\text{V}$
$I_{CC2}$	Standby Supply Current		10	$\mu\text{A}$	$V_{CC} = 5\text{V}$ CLK = (0) $V_{IH} = V_{CC} - 0.2\text{V}$ $V_{IL} = 0.2\text{V}$
$I_{LI}$	Input Leakage Current	-10	10	$\mu\text{A}$	$V_{IN} = 0.4 \text{ to } V_{CC}$
$I_{LO}$	3-State Output Leakage Current in Float	-10	$10^2$	$\mu\text{A}$	$V_{OUT} = 0.4 \text{ to } V_{CC}$

1. Measurements made with outputs floating.

2.  $A_{15}$ - $A_0$ ,  $D_7$ - $D_0$ ,  $MREQ$ ,  $IORQ$ ,  $RD$ , and  $WR$ .

3.  $I_{CC2}$  standby supply current is guaranteed only when the supplied clock is stopped at a low level during  $T_4$  of the machine cycle immediately following the execution of a HALT instruction.

## CAPACITANCE

Symbol	Parameter	Min	Max	Unit
$C_{CLOCK}$	Clock Capacitance		10	pf
$C_{IN}$	Input Capacitance		5	pf
$C_{OUT}$	Output Capacitance		15	pf

$T_A = 25^\circ\text{C}$ ,  $f = 1 \text{ MHz}$ .

Unmeasured pins returned to ground.

# AC CHARACTERISTICS† (Z84C00/CMOS Z80 CPU)

V<sub>cc</sub> = 5.0V ± 10%, unless otherwise specified

No	Symbol	Parameter	Z84C0004**		Z84C0006		Z84C0008		Z84C0010		Z84C0020[1]		Unit	Note
			Min	Max	Min	Max	Min	Max	Min	Max	Min	Max		
1	TcC	Clock Cycle time	250*	DC	162*	DC	125*	DC	100*	DC	50*	DC	nS	
2	TwCh	Clock Pulse width (high)	110	DC	65	DC	55	DC	40	DC	20	DC	nS	
3	TwCl	Clock Pulse width (low)	110	DC	65	DC	55	DC	40	DC	20	DC	nS	
4	TfC	Clock Fall time		30		20		10		10		10	nS	
5	TrC	Clock Rise time		30		20		10		10		10	nS	
6	TdCr(A)	Address valid from Clock Rise		110		90		80		65		57	nS	[2]
7	TdA(MREQf)	Address valid to /MREQ Fall	65*		35*		20*		5*		-15*		nS	
8	TdCl(MREQf)	Clock Fall to /MREQ Fall delay		85		70		60		55		40	nS	
9	TdCr(MREQr)	Clock Rise to /MREQ Rise delay		85		70		60		55		40	nS	
10	TwMREQh	/MREQ pulse width (High)	110*		65*		45**		30*		10*		nS	[3]
11	TwMREQl	/MREQ pulse width (low)	220*		132*		100*		75*		25*		nS	[3]
12	TdCl(MERQr)	Clock Fall to /MREQ Rise delay		85		70		60		55		40	nS	
13	TdCl(RDf)	Clock Fall to /RD Fall delay		95		80		70		65		40	nS	
14	TdCr(RDr)	Clock Rise to /RD Rise delay		85		70		60		55		40	nS	
15	TsD(Cr)	Data setup time to Clock Rise	35		30		30		25		12		nS	
16	ThD(RDr)	Data hold time after /RD Rise	0		0		0		0		0		nS	
17	TsWAIT(Cf)	/WAIT setup time to Clock Fall	70		60		50		20		7.5		nS	
18	ThWAIT(Cf)	/WAIT hold time after Clock Fall	10		10		10		10		10		nS	
19	TdCr(M1f)	Clock Rise to /M1 Fall delay		100		80		70		65		45	nS	
20	TdCr(M1r)	Clock Rise to /M1 Rise delay		100		80		70		65		45	nS	
21	TdCr(RFSHf)	Clock Rise to /RFSH Fall delay		130		110		95		80		60	nS	
22	TdCr(RFSHr)	Clock Rise to /RFSH Rise delay		120		100		85		80		60	nS	
23	TdCl(RDr)	Clock Fall to /RD Rise delay		85		70		60		55		40	nS	
24	TdCr(RDf)	Clock Rise to /RD Fall delay		85		70		60		55		40	nS	
25	TsD(Cf)	Data setup to Clock Fall during M2, M3, M4 or M5 cycles	50		40		30		25		12		nS	
26	TdA(IORQf)	Address stable prior to /IORQ Fall	180*		107*		75*		50*		0*		nS	
27	TdCr(IORQf)	Clock Rise to /IORQ Fall delay		75		65		55		50		40	nS	
28	TdCl(IORQr)	Clock Fall to /IORQ Rise delay		85		70		60		55		40	nS	
29	TdD(WRf)Mw	Data stable prior to /WR Fall	80*		22*		5*		40*		-10*		nS	
30	TdCl(WRf)	Clock Fall to /WR Fall delay		80		70		60		55		40	nS	
31	TwWR	/WR pulse width	220*		132*		100*		75*		25*		nS	
32	TdCl(WRr)	Clock Fall to /WR Rise delay		80		70		60		55		40	nS	
33	TdD(WRf)IO	Data stable prior to /WR Fall	-10*		-55*		-55*		-10*		-30*		nS	
34	TdCr(WRf)	Clock Rise to /WR Fall delay		65		60		60		50		40	nS	
35	TdWRr(D)	Data stable from /WR Rise	60*		30*		15*		10*		0*		nS	
36	TdCl(HALT)	Clock Fall to /HALT 'L' or 'H'		300		260		225		90		70	nS	
37	TwNMI	/NMI pulse width	80		60		60		60		60		nS	
38	TsBUSREQ (Cr)	/BUSREQ setup time to Clock Rise	50		50		40		30		15		nS	

\*For clock periods other than the minimums shown, calculate parameters using the table on the following page.

Calculated values above assumed TrC = TtC = 20 ns.

†Units in nanoseconds (ns).

†† For loading ≥ 50 pF. Decrease width by 10 ns for each additional 50 pF.

\*\*4 MHz CMOS Z80 is obsoleted and replaced by 6 MHz

# AC CHARACTERISTICS† (Z84C00/CMOS Z80 CPU; Continued)

V<sub>CC</sub>=5.0V ± 10%, unless otherwise specified

No	Symbol	Parameter	Z84C0004**		Z84C0006		Z84C0008		Z84C0010		Z84C0020[1]		Unit	Note
			Min	Max	Min	Max	Min	Max	Min	Max	Min	Max		
39	ThBUSREQ (Cr)	/BUSREQ hold time after Clock Rise	10		10		10		10		10		nS	
40	TdCr (BUSACKf)	Clock Rise to /BASACK Fall delay		100		90		80		75		40	nS	
41	TdCf (BUSACKr)	Clock Fall to /BASACK Rise delay		100		90		80		75		40	nS	
42	TdCr(Dz)	Clock Rise to Data float delay		90		80		70		65		40	nS	
43	TdCr(CTz)	Clock Rise to Control Outputs Float Delay (/MREQ, /IORQ, /RD and /WR)		80		70		60		65		40	nS	
44	TdCr(Az)	Clock Rise to Address float delay		90		80		70		75		40	nS	
45	TdCTr(A)	Address Hold time from /MREQ, /IORQ, /RD or /WR	80*		35*		20*		20*		0*		nS	
46	TsRESET(Cr)	/RESET to Clock Rise setup time	60		60		45		40		15		nS	
47	ThRESET(Cr)	/RESET to Clock Rise Hold time	10		10		10		10		10		nS	
48	TsINTf(Cr)	/INT Fall to Clock Rise Setup Time	80		70		55		50		15		nS	
49	ThINTr(Cr)	/INT Rise to Clock Rise Hold Time	10		10		10		10		10		nS	
50	TdM1f (IORQf)	/M1 Fall to /IORQ Fall delay	565*		359*		270*		220*		100*		nS	
51	TdCf(IORQf)	/Clock Fall to /IORQ Fall delay	85		70		60		55		45		nS	
52	TdCf(IORQr)	Clock Rise to /IORQ Rise delay	85		70		60		55		45		nS	
53	TdCf(D)	Clock Fall to Data Valid delay	150		130		115		110		75		nS	

## Notes:

\* For Clock periods other than the minimum shown, calculate parameters using the following table.

Calculated values above assumed TrC = TtC = maximum.

\*\* 4 MHz CMOS Z80 is obsoleted and replaced by 6 MHz

[1] Z84C0020 parameters are guaranteed with 50pF load Capacitance.

[2] If Capacitive Load is other than 50pF, please use Figure 1. to calculate the value.

[3] Increasing delay by 10nS for each 50pF increase in loading, 200pF max for data lines, and 100pF for control lines.

## FOOTNOTES TO AC CHARACTERISTICS

No	Symbol	Parameter	Z84C0004**	Z84C0006	Z84C0008	Z84C0010	Z84C0020
1	TcC	TwCh + TwCl + TrC + TtC					
7	TdA(MREQf)	TwCh + TtC	-65	-50	-45	-45	-45
10	TwMREQh	TwCh + TtC	-20	-20	-20	-20	-20
11	TwMREQf	TcC	-30	-30	-25	-25	-25
26	TdA(IORQf)	TcC	-70	-55	-50	-50	-50
29	TdD(WRf)	TcC	-170	-140	-120	-60	-60
31	TwWR	TcC	-30	-30	-25	-25	-25
33	TdD(WRf)	TwCl + TrC	-140	-140	-120	-60	-60
35	TdWRr(D)	TwCl + TrC	-70	-55	-50	-40	-25
45	TdCTr(A)	TwCl + TrC	-50	-50	-45	-30	-30
50	TdM1f(IORQf)	2TcC + TwCh + TtC	-65	-50	-45	-30	-30

AC Test Conditions: V<sub>IH</sub> = 2.0 V  
V<sub>IL</sub> = 0.8 V

V<sub>OH</sub> = 1.5 V  
V<sub>OL</sub> = 1.5 V

V<sub>IHC</sub> = V<sub>CC</sub> - 0.6 V  
V<sub>ILC</sub> = 0.45 V

FLOAT = ±0.5 V

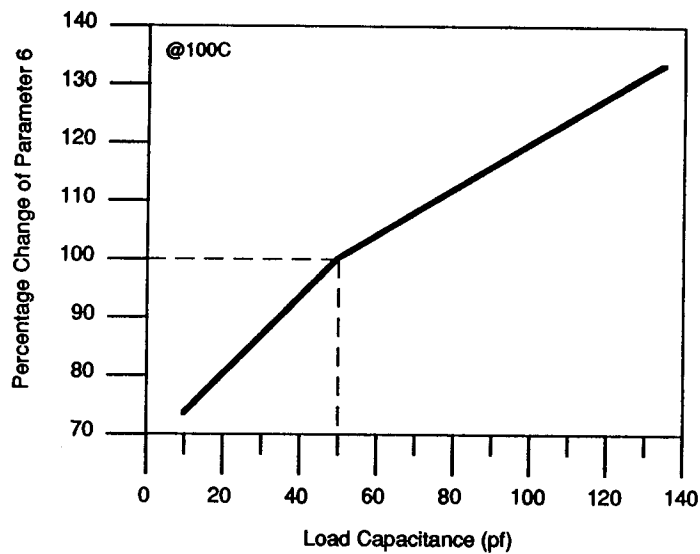


Figure 1. Address Delay Characteristics  
(Parameter 6)

#### DC CHARACTERISTICS (Z8400/NMOS Z80 CPU)

All parameters are tested unless otherwise noted.

Symbol	Parameter	Min	Max	Unit	Test Condition
$V_{ILC}$	Clock Input Low Voltage	-0.3	0.45	V	
$V_{IHC}$	Clock Input High Voltage	$V_{CC} - .6$	$V_{CC} + .3$	V	
$V_{IL}$	Input Low Voltage	-0.3	0.8	V	
$V_{IH}$	Input High Voltage	2.0 <sup>1</sup>	$V_{CC}$	V	
$V_{OL}$	Output Low Voltage		0.4	V	$I_{OL} = 2.0 \text{ mA}$
$V_{OH}$	Output High Voltage	2.4 <sup>1</sup>		V	$I_{OH} = -250 \mu\text{A}$
$I_{CC}$	Power Supply Current		200	mA	Note 3
$I_{LI}$	Input Leakage Current		10	$\mu\text{A}$	$V_{IN} = 0 \text{ to } V_{CC}$
$I_{LO}$	3-State Output Leakage Current in Float	-10	10 <sup>2</sup>	$\mu\text{A}$	$V_{OUT} = 0.4 \text{ to } V_{CC}$

1. For military grade parts, refer to the Z80 Military Electrical Specification.

2.  $A_{15}$ - $A_0$ ,  $D_7$ - $D_0$ ,  $MREQ$ ,  $IORD$ ,  $RD$ , and  $WR$ .

3. Measurements made with outputs floating.

#### CAPACITANCE

Guaranteed by design and characterization.

Symbol	Parameter	Min	Max	Unit
$C_{CLOCK}$	Clock Capacitance		35	pf
$C_{IN}$	Input Capacitance		5	pf
$C_{OUT}$	Output Capacitance		15	pf

#### NOTES:

$T_A = 25^\circ\text{C}$ ,  $f = 1 \text{ MHz}$ .

Unmeasured pins returned to ground.



# AC CHARACTERISTICS† (Z8400/NMOS Z80 CPU)

Number	Symbol	Parameter	Z0840004		Z0840006		Z0840008	
			Min	Max	Min	Max	Min	Max
1	TcC	Clock Cycle Time	250*		162*		125*	
2	TwCh	Clock Pulse Width (High)	110	2000	65	2000	55	2000
3	TwCl	Clock Pulse Width (Low)	110	2000	65	2000	55	2000
4	TfC	Clock Fall Time		30		20		10
5	TrC	Clock Rise Time		30		20		10
6	TdCr(A)	Clock ↑ to Address Valid Delay		110		90		80
7	TdA(MREQf)	Address Valid to $\overline{\text{MREQ}}$ ↓ Delay	65*		35*		20*	
8	TdCf(MREQf)	Clock ↓ to $\overline{\text{MREQ}}$ ↓ Delay		85		70		60
9	TdCr(MREQr)	Clock ↑ to $\overline{\text{MREQ}}$ ↑ Delay		85		70		60
10	TwMREQh	$\overline{\text{MREQ}}$ Pulse Width (High)	110*††		65*††		45*††	
11	TwMREQl	$\overline{\text{MREQ}}$ Pulse Width (Low)	220*††		135*††		100*††	
12	TdCf(MREQr)	Clock ↓ to $\overline{\text{MREQ}}$ ↑ Delay		85		70		60
13	TdCf(RDf)	Clock ↓ to $\overline{\text{RD}}$ ↓ Delay		95		80		70
14	TdCr(RDr)	Clock ↑ to $\overline{\text{RD}}$ ↑ Delay		85		70		60
15	TsD(Cr)	Data Setup Time to Clock ↑	35		30		30	
16	ThD(RDr)	Data Hold Time to $\overline{\text{RD}}$ ↑		0		0		0
17	TsWAIT(Cf)	$\overline{\text{WAIT}}$ Setup Time to Clock ↓	70		60		50	
18	ThWAIT(Cf)	$\overline{\text{WAIT}}$ Hold Time after Clock ↓		0		0		0
19	TdCr(M1f)	Clock ↑ to $\overline{\text{M1}}$ ↓ Delay		100		80		70
20	TdCr(M1r)	Clock ↑ to $\overline{\text{M1}}$ ↑ Delay		100		80		70
21	TdCr(RFSHf)	Clock ↑ to $\overline{\text{RFSH}}$ ↓ Delay		130		110		95
22	TdCr(RFSHr)	Clock ↑ to $\overline{\text{RFSH}}$ ↑ Delay		120		100		85
23	TdCf(RDr)	Clock ↓ to $\overline{\text{RD}}$ ↑ Delay		85		70		60
24	TdCr(RDf)	Clock ↑ to $\overline{\text{RD}}$ ↓ Delay		85		70		60
25	TsD(Cf)	Data Setup to Clock ↓ during M <sub>2</sub> , M <sub>3</sub> , M <sub>4</sub> , or M <sub>5</sub> Cycles	50		40		30	
26	TdA(IORQf)	Address Stable prior to $\overline{\text{IORQ}}$ ↓	180*		110*		75*	
27	TdCr(IORQf)	Clock ↑ to $\overline{\text{IORQ}}$ ↓ Delay		75		65		55
28	TdCf(IORQr)	Clock ↓ to $\overline{\text{IORQ}}$ ↑ Delay		85		70		60
29	TdD(WRf)	Data Stable prior to $\overline{\text{WR}}$ ↓	80*		25*		5*	
30	TdCf(WRf)	Clock ↓ to $\overline{\text{WR}}$ ↓ Delay		80		70		60
31	TwWR	$\overline{\text{WR}}$ Pulse Width	220*		135*		100*	
32	TdCf(WRr)	Clock ↓ to $\overline{\text{WR}}$ ↑ Delay		80		70		60
33	TdD(WRf)	Data Stable prior to $\overline{\text{WR}}$ ↓	-10*		-55*		55*	
34	TdCr(WRf)	Clock ↑ to $\overline{\text{WR}}$ ↓ Delay		65		60		55
35	TdWRr(D)	Data Stable from $\overline{\text{WR}}$ ↑	60*		30*		15*	
36	TdCf(HALT)	Clock ↓ to $\overline{\text{HALT}}$ ↑ or ↓		300		260		225
37	TwNMI	$\overline{\text{NMI}}$ Pulse Width	80		70		60*	
38	TsBUSREQ(Cr)	$\overline{\text{BUSREQ}}$ Setup Time to Clock ↑	50		50		40	

\* For clock periods other than the minimums shown, calculate parameters using the table on the following page. Calculated values above assumed TrC = TfC = 20 ns.

† Units in nanoseconds (ns).

†† For loading ≥ 50 pF., Decrease width by 10 ns for each additional 50 pF.

# AC CHARACTERISTICS† (Z8400/NMOS Z80 CPU; Continued)

Number	Symbol	Parameter	Z0840004		Z0840006		Z0840008	
			Min	Max	Min	Max	Min	Max
39	ThBUSREQ(Cr)	BUSREQ Hold Time after Clock ↑	0		0		0	
40	TdCr(BUSACKf)	Clock ↑ to BUSACK ↓ Delay		100		90		80
41	TdCl(BUSACKr)	Clock ↓ to BUSACK ↑ Delay		100		90		80
42	TdCr(Dz)	Clock ↑ to Data Float Delay		90		80		70
43	TdCr(CTz)	Clock ↑ to Control Outputs Float Delay (MREQ, IORQ, RD, and WR)		80		70		60
44	TdCr(Az)	Clock ↑ to Address Float Delay		90		80		70
45	TdCTr(A)	MREQ ↑, IORQ ↑, RD ↑, and WR ↑ to Address Hold Time	80*		35*		20*	
46	TsRESET(Cr)	RESET to Clock ↑ Setup Time	60		60		45	
47	ThRESET(Cr)	RESET to Clock ↑ Hold Time		0		0		0
48	TsINTf(Cr)	INT to Clock ↑ Setup Time	80		70		55	
49	ThINTr(Cr)	INT to Clock ↑ Hold Time		0		0		0
50	TdM1f(IORQf)	M1 ↓ to IORQ ↓ Delay	565*		365*		270*	
51	TdCl(IORQf)	Clock ↓ to IORQ ↓ Delay		85		70		60
52	TdCl(IORQr)	Clock ↑ to IORQ ↑ Delay		85		70		60
53	TdCr(D)	Clock ↓ to Data Valid Delay		150		130		115

\*For clock periods other than the minimums shown, calculate parameters using the following table. Calculated values above assumed TrC = TrC = 20 ns.

†Units in nanoseconds (ns).

## FOOTNOTES TO AC CHARACTERISTICS

Number	Symbol	General Parameter	Z0840004	Z0840006	Z0840008
1	TcC	TwCh + TwCl + TrC + TrC			
7	TdA(MREQf)	TwCh + TrC	- 65	- 50	- 45
10	TwMREQh	TwCh + TrC	- 20	- 20	- 20
11	TwMREQl	TcC	- 30	- 30	- 25
26	TdA(IORQf)	TcC	- 70	- 55	- 50
29	TdD(WRf)	TcC	- 170	- 140	- 120
31	TwWR	TcC	- 30	- 30	- 25
33	TdD(WRf)	TwCl + TrC	- 140	- 140	- 120
35	TdWRr(D)	TwCl + TrC	- 70	- 55	- 50
45	TdCTr(A)	TwCl + TrC	- 50	- 50	- 45
50	TdM1f(IORQf)	2TcC + TwCh + TrC	- 65	- 50	- 45

AC Test Conditions:

V<sub>IH</sub> = 2.0 V

V<sub>IL</sub> = 0.8 V

V<sub>IHC</sub> = V<sub>CC</sub> - 0.6 V

V<sub>ILC</sub> = 0.45 V

V<sub>OH</sub> = 1.5 V

V<sub>OL</sub> = 1.5 V

FLOAT = ±0.5 V

# Customer Support

For answers to technical questions about the product, documentation, or any other issues with Zilog's offerings, please visit Zilog's Knowledge Base at <http://www.zilog.com/kb>.

For any comments, detail technical questions, or reporting problems, please visit Zilog's Technical Support at <http://support.zilog.com>.